UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/665,305 | 09/22/2003 | Thomas Goering | 11884-400201 | 5450 |

53000 7590 09/12/2009
KENYON & KENYON LLP
1500 K STREET N.W.
WASHINGTON, DC 20005

| EXAMINER | |
|---|---|
| DESAI, RACHNA SINGH | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2176 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 09/12/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

*Ex parte* THOMAS GOERING

Appeal 2009-002814
Application 10/665,305
Technology Center 2100

Decided: September 14, 2009

Before JOHN A. JEFFERY, ST. JOHN COURTENAY III, and STEPHEN C. SIU, *Administrative Patent Judges*.

JEFFERY, *Administrative Patent Judge*.

## DECISION ON APPEAL

Appellant appeals under 35 U.S.C. § 134(a) from the Examiner's rejection of claims 1-18. We have jurisdiction under 35 U.S.C. § 6(b), and we heard the appeal on September 9, 2009. We affirm.

## STATEMENT OF THE CASE

Appellant's invention generates output modules in a form-based runtime environment. In one implementation, a form manager (1) receives an indication that a reusable form element has changed; (2) determines which output modules are affected by the changed form element; and (3) invalidates the affected output modules. If a requested output module has been invalidated, it is regenerated.[1] Claim 9 is illustrative with the key disputed limitations emphasized:

> 9. A computer-implemented method for generating output modules in a form-based application runtime environment, comprising:
>
> receiving an indication that a reusable form element has been changed;
>
> determining which output modules from a set of output modules are affected by the changed form element;
>
> *invalidating the affected output modules;*
>
> receiving a request for an output module from the set of output modules; and
>
> *regenerating the requested output module if the requested output module has been invalidated.*

The Examiner relies on the following as evidence of unpatentability:

| | | |
|---|---|---|
| Hitchcock | US 2005/0080756 A1 | Apr. 14, 2005<br>(eff. filed June 4, 1998) |

The Examiner rejected claims 1-18 under 35 U.S.C. § 103(a) as unpatentable over Hitchcock. Ans. 3-13.

[1] *See generally* Abstract; Spec. ¶ 0010.

Rather than repeat the arguments of Appellant or the Examiner, we refer to the Briefs and the Answer[2] for their respective details. In this decision, we have considered only those arguments actually made by Appellant. Arguments which Appellant could have made but did not make in the Briefs have not been considered and are deemed to be waived. *See* 37 C.F.R. § 41.37(c)(1)(vii).

Regarding representative claim 9,[3] the Examiner finds that Hitchcock discloses all of the claimed subject matter except for expressly stating that output modules are invalidated. The Examiner, however, cites Hitchcock's ability to automatically (1) populate previously-entered data in form fields, and (2) update this data in accordance with user-entered changes. Based on this capability, the Examiner contends that since Hitchcock's system acknowledges that the older data (i.e., before the update) is no longer valid, it therefore "invalidates" the output modules that would present the now-invalid data to the user (i.e., by automatically populating the data in the form's fields). Ans. 6, 14-16. The Examiner further contends that these invalidated output modules are regenerated to present the form with the new data. Ans. 17.

[2] Throughout this opinion, we refer to (1) the Appeal Brief filed March 20, 2008; (2) the Examiner's Answer mailed April 3, 2008; and (3) the Reply Brief filed June 3, 2008.

[3] Appellant argues claims 1-18 together as a group. *See* App. Br. 4-7. Accordingly, we select independent claim 9 as representative. *See* 37 C.F.R. § 41.37(c)(1)(vii).

Appellant argues that Hitchcock does not (1) invalidate output modules, and (2) regenerate invalidated output modules as claimed. According to Appellant, Hitchcock teaches away from invaliding output modules since the applicant database can be extended to include new attributes without making any changes to the forms engine program. App. Br. 4-5.

Appellant adds that the Examiner incorrectly equates the claimed "output module" with form data. Appellant emphasizes that an "output module" is an executable module of a software system that can be called by other applications—a feature distinct from Hitchcock's current forms which are said to be data records. App. Br. 5; Reply Br. 4-6.

Appellant also contends that Hitchcock teaches away from regenerating output modules as claimed since Hitchcock's forms engine—which Appellant indicates corresponds to an "output module"—is extensible without programming. App. Br. 6.

The issues before us, then, are as follows:

## ISSUES

Under § 103, has Appellant shown that the Examiner erred in rejecting claim 9 by finding that Hitchcock teaches or suggests:

(a) invalidating output modules affected by a changed form element, and

(b) regenerating invalidated output modules as claimed?

## FINDINGS OF FACT

The record supports the following findings of fact (FF) by a preponderance of the evidence:

### *Hitchcock*

1. Hitchcock discloses a universal forms engine that allows data sharing between customizable online forms, such as college admissions applications. After the applicant completes an application for one institution, the data is saved in a database and automatically populates fields in subsequent application forms. Hitchcock, Abstract.

2. To this end, forms engine 104 (1) generates a customized application form based on an application description in an application data file 108; (2) retrieves user data that was entered in previous applications and stored in applicant database 62; and (3) merges this data into the current application which is returned to the user as an HTML form. The applicant then enters any requested information that was not automatically inserted from the database. Hitchcock, ¶¶ 0048-49; Fig. 13.

3. When the applicant subsequently applies to a different institution, a new customized application is presented to the applicant. Information that was entered into previously-submitted applications is retrieved from the database and presented to the applicant as populated fields of the new application. Hitchcock, ¶ 0056.

4. The applicant can change the values in a pre-populated field if desired, and the new values are saved for use in subsequent applications. Hitchcock, ¶ 0056.

5. The application data file describes the format of each application, and the forms engine displays information from the database in the format prescribed by the application data file. Hitchcock, ¶ 0064.

6. The application data file is a specially-formatted text file that is a series of "directives" and optional arguments which the forms engine parses to build the HTML form and to merge in user data. Hitchcock, ¶ 0080.

7. The directives are interpreted by means of a look-up in a data structure that stores the directive interpretations. For example, the forms engine can interpret a line in the application data file indicated as "SS_NUM" to (1) display a text box with an associated label "Enter Your Social Security Number," and (2) put the previously-supplied value for the social security number (stored in the User Attribute Table) into the text box. Hitchcock, ¶ 0080.

8. The application data file can supply arguments to directives that can, among other things, instruct the forms engine to override default values to customize the data entry format. Hitchcock, ¶ 0080.

9. When an applicant enters information on an application page and posts the form to the server, the entered information is stored in the User Attribute Table after validation. Hitchcock, ¶ 0071.

10. The User Attribute Table always includes the latest information that an applicant entered and is used to supply information for new applications. Hitchcock, ¶ 0072.

11. The applicant database can be extended to include new attributes without making any changes to the forms engine program. Also, the appearance of the application for each institution can be changed by changing its application description file, without reprogramming the forms engine. Hitchcock, ¶ 0065.

### *Appellant's Disclosure*

12. Appellant's Specification notes that once a created form is activated, the runtime system "automatically generates an ABAP function module (i.e., form output module) that can subsequently be called by an application program . . . . The form output module processes the imported application-specific data and its form description data for presentation via spool (printer), fax, e-mail, web browser, etc." Spec. ¶ 0006.

## PRINCIPLES OF LAW

In rejecting claims under 35 U.S.C. § 103, it is incumbent upon the Examiner to establish a factual basis to support the legal conclusion of obviousness. *See In re Fine*, 837 F.2d 1071, 1073 (Fed. Cir. 1988). If the Examiner's burden is met, the burden then shifts to the Appellant to overcome the prima facie case with argument and/or evidence. Obviousness is then determined on the basis of the evidence as a whole and the relative persuasiveness of the arguments. *See In re Oetiker*, 977 F.2d 1443, 1445 (Fed. Cir. 1992).

## ANALYSIS

We begin by construing the term "output module" for it is a critical limitation in claim 9 on which this appeal hinges. Appellant emphasizes that "output modules" must be executable and capable of being called by other applications. App. Br. 5; Reply Br. 4-6. This construction is consistent with Appellant's usage of the term in the Specification which equates a "form output module" to a function module that not only can be called by an application program, but also processes imported application-specific data and form description data for presentation. FF 12. Based on these properties, we interpret an "output module" as more than mere form data, but rather functional computer code that can be interpreted or executed.

But even under this interpretation, we agree with the Examiner that Hitchcock reasonably teaches or suggests (1) "invalidating" output modules, and (2) "regenerating" invalidated output modules as claimed. Hitchcock's forms engine customizes application forms by, among other things, (1) retrieving previously-entered user data stored in a database, and (2) merging this data into the current application which is returned to the user as an HTML form. FF 2. Essentially, the system populates fields of application forms with previously-submitted information. FF 3.

The applicant, however, can change the values in a pre-populated field if desired, and the new values are saved for use in subsequent applications. FF 4. At this point, the newer values—not the older values—would be populated in the application fields. *See id.* While the *data itself* that is presented to the user as a populated field on an application form may be non-functional, the same cannot be said for the *underlying code* that is used to automatically generate this updated presentation.

And in this sense, we agree with the Examiner that this change would cause Hitchcock's forms engine to "invalidate" the module used to automatically populate the fields with the older, outdated (and now invalid) information in favor of the most recent (and valid) information. Since skilled artisans would recognize that the underlying code would be changed at least minimally to achieve this end, this "output module" would have therefore been "regenerated" to reflect these changes.

Hitchcock all but confirms this point. Hitchcock's forms engine relies extensively on information from an application data file to display information from the database accordingly. FF 5. Significantly, the application data file is a specially-formatted text file including a series of "directives" and arguments that the forms engine parses to build the HTML form and to merge in user data. FF 6. The forms engine interprets these directives to perform associated display functions involving user data, including, among other things, (1) displaying text boxes associated with user information, and (2) filling in these text boxes with previously-entered values stored in the User Attribute Table. FF 7-9.

As such, we find that the recited "output module" is not limited to just the forms engine itself as Appellant seems to suggest (*see* App. Br. 6), but also includes the code used to generate a particular form—code that varies, at least in part, based on the stored user data. *See* FF 5-10.

This process of automatically building a form based on the most recent information gleaned from interpreting directives and arguments—code that relies on particular information from the User Attribute Table (FF 7)—would at least implicitly (1) "invalidate" the underlying code associated with the older information responsive to a changed "form element" (i.e., the field and its corresponding information), and (2) "regenerate" the code based on new information. Although Hitchcock indicates that various aspects of the system can be modified without reprogramming the forms engine (FF 11), nothing in claim 9 precludes the implicit "invalidation" and "regeneration" that occurs when the output module used to generate a particular display with outdated data in Hitchcock is changed to generate a different display with updated data.

For the foregoing reasons, Appellant has not persuaded us of error in the Examiner's rejection of representative claim 9. Therefore, we will sustain the Examiner's rejection of that claim, and claims 1-8 and 10-18 which fall with claim 9.

## CONCLUSION

Appellant has not shown that the Examiner erred in rejecting claims 1-18 under § 103.

## ORDER

The Examiner's decision rejecting claims 1-18 is affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

## AFFIRMED

pgc

KENYON & KENYON LLP
1500 K STREET N.W.
WASHINGTON DC 20005